Study on Query Processing Mechanism of OGSA-DQP

TAN Yue-sheng [1], WU Zhi[2], WANG Jing-yu[1]

[1](Information and Network Center of Inner Mongolia University of Science &Technology,Inner Mongolia Bao Tou 014010)

[2](Outside Capital Construction of Lu Dong University , Shan Dong Yan Tai 264025)

***Abstract*—OGSA-DQP is A Service-Based Distributed Query Processing system in the Grid，It is concerned with providing a an approach to deliver High level Data Access and Integration services which are needed if data-intensive distributed applications running on heterogeneous platforms are to benefit from the Grid. It provides a uniform virtual data view above multiple relational database in the grid, Grid users can issue their queries based on the virtual view. Firstly, the architecture of OGSA-DQP was described. Then, the internal query and optimization mechanism of OGSA-DQP was analyzed. The query performance was evaluated for finding the bottlenecks of query performance and improving the query response time.**

Keywords- DQP;Grid; OGSA;

I. INTRODUCTION

Data resources of grid environment are stored in the network in the form of decentralized and distributed form.It takes on heterogeneous, decentralized and autonomous features. Grid-based query is a kind of distributed queries. The data which users need are stored in different locations or databases, especially when the applications in grid refer to a large number of data with complex structure and semantics data. such as in the grid applications, the operation of distributed query processing on the virtual database which is constructed dynamically and each database on the grid node may invovle only a part of the necessary information. Meanwhile, some operation of data query in relation to the heterogeneity of the databases and the cost of communication that will bring about some problems in the grid. Commonly, in the large-scaled, decentralized and heterogeneous environment, distributed query system can not meet our needs for processing or accessing the data in the grid. Therefore, we need a component which can integrate and access a large amount of distributed data. The OGSA-DQP[1] is a middleware in the grid environment for data integration. It is an essentially high performance, distributed data flow engine, which supports the integration and query for distributed and heterogeneous data,It also supports the query optimization and evaluation.

II. OGSA AND OGSA-DAI

Open Grid Services Architecture(OGSA) is called the next generation of grid architecture. Based on the original five-tier hourglass model, it was proposed by combining with Web Service technology. The core idea of OGSA is “Service-centered”, and it takes everything as service. Commonly, Web Service often deal with static service , but in the environment of grid, most of the services are temporary, such as the computing task execution,etc. Considering the features of network environment, OGSA present a concept of “Grid Service” based on Web Service.Grid Service is a kind of Web Services, which provides a set of interfaces to solve the problem related to the temporary services, such as the service discovery, dynamic service creation , service lifecycle Management, etc.

OGSA-DAI is a middleware which provides a unified service interface to access the grid and integrate different data sources. the different and heterogeneous data sources are logically regarded as a whole by means of OGSA-DAI interface. OGSA-DAI grid services provide some basic operations to execute complicated task, for instance,data union and distributed query in the virtual organizations. Moreover, many technical details , for example, database driver, data format ,transport mechanism, are hidden in the OGSA-DAI grid services.The core of NAC is security control of network access, which includes authentication server, security policy server, as well as third-party software such as anit-virus server and system patch server and so on . It checks the security status of host before access network. and applies different access control policy according to results. If the results can’t accord with network security standards, the host will be separated from network and forced to update virus database, install system patches and so on. On the premise of host security with self-defense capability, it is reasonably to control the user’s network behavior and improve the whole network security and defense capability in order to implement the aim of active defense.

III. OGSA-DQP ARCHITECTURE AND QUERY MECHANISM

OGSA-DQP is an essentially high performance, distributed data flow engine based on OGSA-DAI service architecture. It relies on the grid resource extraction of service-oriented and assumes that the data sources can be accessed via the service interface,he architecture of OGSA-DQP is shown in Figure 1.

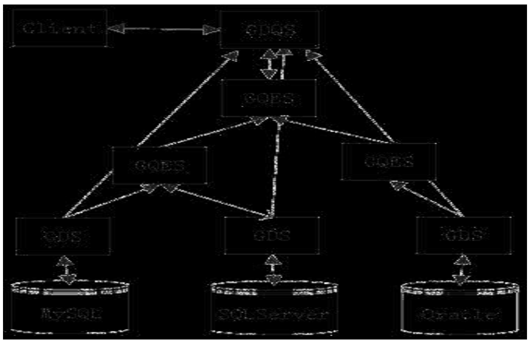

Figure 1. the architecture of OGSA-DQP

OGSA-DQP extends OGSA-DAI with the following two services :

This work has been supported by nature science fundment project of inner mongolia(No:20080404ms0903);

- GDQS (Grid Distributed Query Services).

It is a interface to interact with the clients. When we need to create a new SQL query, GDQS will collect necessary metadata and the information of computing resources, and then compile, optimize, divide, schedule and execute the distributed query plan over the grid nodes.

- GQES (Grid Query Evaluation Service).

GQES is used to execute the query plan created by the compiler, optimizer and the scheduler. Each evaluator is in charge of the query execution plan which is assigned.

OGSA-DQP is implemented by means of middlewares [2], which are taken as the OGSA-DAI wrappers. As is shown in Figure 2, OGSA-DQP is based on OGSA-DAI which is responsible for the interaction with DBMS to complete the related data operation .

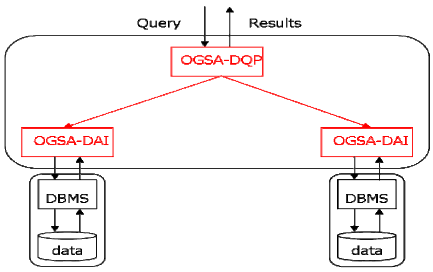

Figure 2. the implement method of OGSA-DQP

A. OGSA-DQP Query Mechanism

the mechanism of distributed query processing is shown in Figure 3 ,and it indicates two phases of optimization which include the optimization of single-node and multi-node.

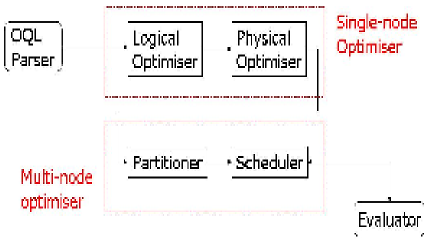

Figure 3. OGSA distributed query mechanism

When a query is submitted, firstly it goes through a parser which checks the type of metadata information of the resources, and the results are expressed as a tree. Afterward, the parser submits the tree to the logical optimizer to generate a logical plan. This logical plan is expressed as another tree whose leaf-nodes are related to the query operation.

Followed by physical optimization, a logical plan is transformed into a physical plan expressed as a tree. Logical operators may related to multiple physical operators, and the lowest cost plan is chosen and executed via a cost-based ranking model. In the mutli-node optimization phase, a partitioner breaks down the single-node execution plan into partitions and a scheduler assigns the partitions to execution nodes. During this phase , the optimiser considers parallelising certain operators in order to speed up the query execution, moreover, a query plan may be divided into multiple sub-plans.

B. OGSA-DQP query optimization

A representative example of the query optimization is as follow: select p.proteinId, blast (p.sequence) from p in protein, t in goTerm where t.termId = \'GO:0005942' and p.proteinId = t.proteinId .

This query requests to access two databases invovling Go and GIMS. Protein is a table of GIMS[3], and goTerm is a database object of Go. This query calls a BLAST sequence similarity program which shows the protein sequences. The query result is the protein sequence ID and similarity result.

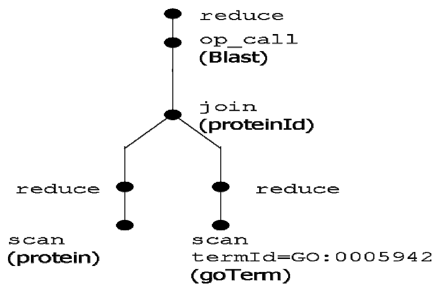

Figure 4. the logic optimization of example query

Figure 4 depicts the logical optimization of the example query. The logical optimization implements all kinds of query transform and generate multiple equivalent plans, such as the join of multi-select operation,etc. The query plan is expressed using a logical algebra, moreover,the aim of Reduce operator in this figure is to bind a input metadata stream with a variable or a variable set . Join is a relevance operator,it will join these metadata sets when certain conditions are fufilled.

Physical optimizer is to transform the optimized logical expression to physical algebra,namely plan is expressed using a physcial algebra ,and the logical operators are replaced by the physical operators to execute each operation in the logical plans and choose a suitable plan of cost-based ranking model. It is shown in Figure 5.

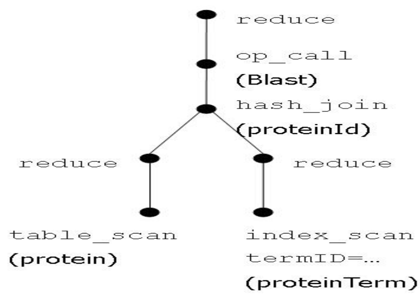

Figure 5. the physical optimization of example query

Following the physical optimization is the query partition phase, in this phase, plan is expressed in a parallel algebra,and the query partition firstly identifies whether an operator needs to divide its input data stream by a specific property, and the query partition inserts a parallel operator which is converted to parallel algebra (parallel algebra=physical algebra + exchange),

as is shown in Figure 6, exchange operator is mainly responsible for data partition and internal processing/service communication, etc. for every exchange operator, we need to define a data partition strategy. Afterward, the query partition must check whether the data need to re-partitioning or be exchanged between the processors,and so on . in addition,exchange operators are commonly placed where data movement may be required.

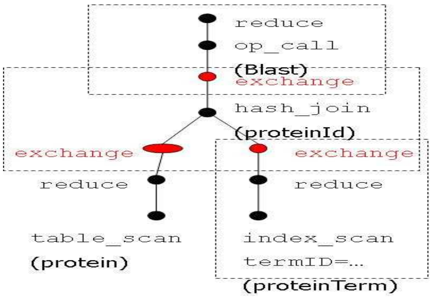

Figure 6. the partition of example query

In the scheduling phase, as shown in Figure 7, the query partitions expressed by decorating parallel algebra expression are allocated to Grid nodes and these partitions may be merged during scheduling. And besides,Heuristic algorithm Considering Memory use, network costs is executed in this phase.

As to the hash_join operator in the Figure 7, query scheduler will try to ensure that there are enough available memory to allocate for the construction of HASH table, and maybe it will combine multiple plans into one plan.

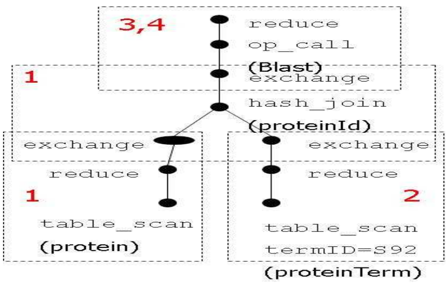

Figure 7. the schedule of example query

In the query evaluation phase, it is done by the Evaluator factory, which is a permanent service. Fistly, GQESs create for partitions as required and then partitions are sent to GQESs. Secondly, partitions are evaluated using iterator model,and then the partitioned and scheduled sub-plans are executed by dynamically creating and deploying the evaluator instance. Thirdly, these execution results are conveyed to client .

IV. QUERY PERFORMANCE EVALUATION AND ANALYSIS

A. Experimental environment

System software and data sources used in the experiment: OGSA-DAI 7.0 and OGSA-DQP 3.0, data sources use two representative system demo table stored in the mySQL of OGSA-DQP system: protein_goterm and protein_interaction. In this experiment , the mySQL database is stored in six different hosts, which are connected via 100Mb LAN.

The query test statements[4] are as follow:

Query 1: select * from protein_goterm or select * from protein_interaction

Query 2: select I.ORF2 from protein_goterm as p, protein_interaction as I where p.ORF=i.ORF1

B. Results of the experiment

In the query performance test, we mainly computed the time consumption when using the JDBC[5][6] and OGSA-DQP join to execute different quantities of query requirement by query 1, and finally the average value of multiple measurements are adopted as shown in Figure 8:

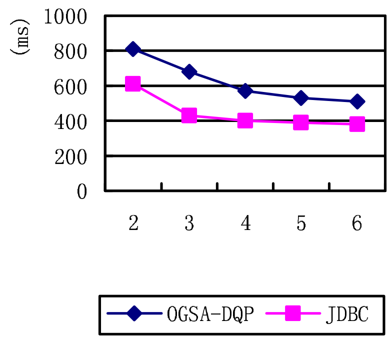

Figure 8. the exec times of query 1 in different query conditions

From Figure 8, wiht the increasing of the number of query condition, the returned data records becomes more less. The system Processing time was on the decrease. it shows that the return of results consume most of the query time, and especially the OGSA-DQP query was executed . The experiment datas indicates that when we execute the non-join query, most of the time is consumed on the data transmission. we found that OGSA-DQP consumed more time to process contrasting with the JDBC query .

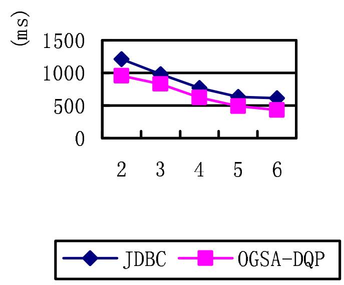

Figure 9. the exec times of query 2 in different node numbers

But when we executed the query with the connect condition, as is shown in Figure 9, the computing time took up most of the total processing time. The more involved nodes, OGSA-DQP became more and more efficient.

V. CONCLUSION

This paper discussed in detail about the distributed query processing mechanism based on service in the grid environment. We compared the performance of query of JDBC and OGSA-DQP under the different condition and different nubmer of nodes. Although OGSA-DQP's performance in the jion query is more better,but its query efficiency is still a bottleneck. At the same time, the task exection of OGSA-DQP is very complex, especially when it is applied to the service-oriented grid. Because of many things are unpredictable in the grid , so maybe more bottleneck will be found.in the future we will spend more time to improve the query efficiency of OGSA-DQP .

REFERENCES

[1] J. Smith, A. Gounaris, P. Watson, et al. Distributed Query Processing on the Grid[A]. In Proc.Grid Computing 2002[C],Berlin: Springer-Verlag, 2002. 279-290.

[2] M. N. Alpdemir, A. Mukherjee, N.W. Paton,et al. Service-based distributed querying on the grid[A]. In the Proceedings of the First International Conference on Service Oriented Computing[C], Trento: Springer-Verlag, 2003. 467-482.

[3] M. Corell, N. W. Paton, S. Wu, et al. GIMS-A Data Warehouse for Storage and Analysis of Genome Sequence and Functional Data[A]. In Proc.2nd IEEE Symposium on BIBE[C], Washington, DC:IEEE Press,2001. 15-22.

[4] Alpdemir, M.N. Gounaris, A. Mukherjee, et al. Experience on Performance Evaluation with OGSA-DQP[A]. In Proceedings of the UK e-Science All Hands Meeting 2005[C], Nottingham: Springer-Verlag, 2005. 453-461.

[5] shi ke,lin hai hua,anyquery:the service-based distributed query processing system in the grid[J], Mini-Micro Systems,2006,27(8):1433-1437.

[6] yuan ke qing,luo si wei,DQP: a distributed query processor on the grid[J],computer technology and development,2006,16(3):92-94.